A Confidence Weighted Real-Time Depth Filter for 3D Reconstruction

Zhenzhou Shao, Zhiping Shi, Ying Qu, Yong Guan, Hongxing Wei, Jindong Tan

A Confidence Weighted Real-Time Depth Filter for 3D Reconstruction

Zhenzhou Shao[1], Zhiping Shi[1⋆], Ying Qu[2⋆],
Yong Guan[1], Hongxing Wei[3] and Jindong Tan[4]

[1] Light-weight Industrial Robot and Safety Verification Lab,
College of Information Engineering, Capital Normal University, China
[2] Department of Electrical Engineering and Computer Science,
The University of Tennessee, Knoxville, USA
[3] School of Mechanical Engineering and Automation, Beihang University, China
[4] Department of Mechanical, Aerospace, and Biomedical Engineering,
The University of Tennessee, Knoxville, USA.
{zshao,shizp,guanyong}@cnu.edu.cn
{yqu3,tan}@utk.edu
weihongxing@buaa.edu.cn

Abstract. 3D reconstruction is an important technique in the environmental perception and rehabilitation process. With the help of active depth-aware sensors, such as Kinect from Microsoft and SwissRanger, the depth map can be captured at the video frame rate together with color information to enable the real-time reconstruction. Particularly, it features prominently in the activity recognition and remote rehabilitation. Unfortunately, the coarseness of the depth map make it difficult to extract the detailed information in 3D reconstruction of the scene and tracking of thin objects. Especially, geometric distortions occur around the edge of an object. Therefore, this paper presents a confidence weighted real-time depth filter for the edge recovery to reduce the extra artifacts due to the uncertainty of each depth measurement. Also the intensity of depth map is taken into account to optimize the weighting term in the algorithm. Moreover, the GPU implementation guarantees the high computational efficiency for the real-time applications. Experimental results are shown to illustrate the performance of the proposed method by the comparisons with the traditional methods.

Keywords: Depth sensor, depth filter, image processing, 3D reconstruction

1 Introduction

3D reconstruction technique is usually used for the rehabilitation purposes, such as activity recognition in remote rehabilitation [1], facial muscle and tongue movement capture in speech recovery sessions [2], joint kinematics supervision [3] and so on. In

⋆ Corresponding author. This paper is supported by Beijing Advanced Innovation Center for Imaging Technology, Development and application of domestic robot embedded real-time operating system (No. 2015BAF13B01) and Training young backbone talents personal projects (No. 2014000020124G135).

recent years, the emergence of depth-aware sensors, such as Flash Lidar [4], Time-of-Flight (ToF) [5] and Kinect from Microsoft, provides a potential solution for real-time 3D reconstruction. This kind of sensor can capture the depth image at video frame rate, and is quite suitable for 3D rendering of the dynamic scenes by integrating with the color camera. It can be used for environmental perception, autonomous navigation, data visualization, robotics and 3D entertainment. Detailed applications based on depth sensors are reviewed in [6].

However, the depth map generated by depth sensors is interfered by the random noise and systematic errors. Especially, some geometric distortions with invalid depth measurements are introduced around the edge of an object. The coarseness of the depth measurements make it difficult to extract the detailed information for further processing such as segmentation, measurement and human pose estimation. Therefore, edge recovery is necessary to guarantee the high fidelity of final 3D rendering. Another problem is the real-time requirement for some applications. The effectiveness and computational cost of the filter implementation have to be considered.

There exist some algorithms to reduce the noise level of the depth map. A few denoising approaches are proposed based on a bilateral filter, which is an edge-preserving and noise reducing smoothing filter[7]. In this paper, we call it standard bilateral filter to distinguish from the algorithm below. In [8], joint/cross bilateral filter (JBF) is presented using the color information to enhance the depth map. It assumes that the depth edges are consistent with the color edges. Unfortunately, the guide of color information runs the risk of the introduction of the new artifacts from the color image. Therefore, Chan et al. [9] proposed a noise-aware filter incorporating the depth intensity to balance the influence from the color image. However, the introduction of the noise-contained depth intensity also brings extra uncertainty. Another alternative way to denoise the depth map is called Non-Local Means (NL-Means) filter [10], which is the extension of bilateral filter. Considering the noise in the depth map, patches surrounding the filtered pixel and neighborhood are taken into account instead of comparing the single pixel value. Although the accuracy is higher, it is not suitable for the real-time applications due to the heavy overhead.

This paper presents a real-time Confidence Weighted Depth Filter (CWDF) for 3D reconstruction to deal with the geometric distortions around the edges of objects. Following the concept of the noise-aware filter, a confidence map of the depth sensor is estimated as the weight of the depth intensity to reduce the influence of the noise from depth measurements, which also solves the problem NL-means filter can deal with. To speed up the filter implementation, the filter is decomposed into a number of constant time spatial filters. And with the GPU support, the proposed algorithm can be performed in a real-time manner.

2 CWDF: Confidence weighted depth filter

Depth measurements are interfered by the random noise and artifacts, so that the neighbors for averaging the gray-scale depth produce undesirable artifacts near strong discontinuities especially. In order to reduce the effect on the weighted result, a confidence weighted depth filter is proposed as a modified noise-aware depth filter, where con-

fidence estimation $C(q)$ is introduced into the computation of the weighting term, as shown in Eq. 1.

$$\begin{aligned} I^C(p) = & \frac{1}{K(p)} \sum_{q \in \Omega} f_s(\|p-q\|)[\alpha(\triangle\Omega) f_r(\|\tilde{I}(p)-\tilde{I}(q)\|) \\ & + (1-\alpha(\triangle\Omega))C(q) f_{r'}(\|I(p)-I(q)\|)]I(q) \end{aligned} \tag{1}$$

where $f_s, f_r, f_{r'}$ are all Gaussian kernels. Compared with JBF, $f_{r'}$ is a new range term. $K(p)$ is a normalizing factor. Confidence measurement is denoted by $C(q)$, which will be estimated using the method in Sect. 2.1. And $C(q)$ features the proposed method to optimize the noise-aware filter in this paper. $\alpha(\triangle\Omega)$ is used to balance the influence of depth and color intensity, as shown in Eq. 2.

$$\alpha(\triangle\Omega) = \frac{1}{1+e^{-\varepsilon(\triangle\Omega-\tau)}} \tag{2}$$

where the parameters ε and τ are chosen to adjust the rate of change in transition area and the blending influence at the minimal min-max difference, respectively. These values depend on the characteristics of the sensor, and can be obtained by the empirical experiments. $\triangle\Omega$ is the difference between the maximum and minimum gray-scale value in the neighbors Ω of the pixel in depth map. When α is higher, the filter behaves like a standard bilateral filter with the $C(q)$.

2.1 Confidence estimation

In this paper, Kinect is chosen to construct the depth map based on the light coding technique. The emissive light is organized using the speckle pattern, which is projected onto the surface of an object. The speckle pattern can change along with the different distance. Through comparing with the reference pattern, the depth map can be constructed.

Confidence estimation is mainly used to determine the reliability of each value in the depth map. In this paper, the confidence measurement is taken into account following the truth that depth values that are more reliable should make more contributions to the final result. To estimate the confidence, absolute difference is employed as the cost. In [11], the depth d_0 with the lowest cost is considered as the true value, although the depth perturbed by the noise and artifacts. We assume the cost of each pixel is subjected to Gaussian distribution. Let $c(d,x)$ be the cost for depth d at pixel x. The probability is proportional to $e^{-(c(d,x)-c(d_0,x))^2/\sigma^2}$, where d_0 is the ground truth for a specific depth, and σ^2 denotes the strength of the noise.

For each specific depth d_i, the confidence $C'(x,d_i)$ is defined as the inverse of the sum of these probabilities for all possible depths. In order to estimate the confidence better, $C'(x,d_i)$ is calculated with the N truth values. Then the final confidence $C(x)$ can be derived by averaging the estimations at the different depth, as shown in Eq. (3).

$$
\begin{aligned}
C'(x,d_i) &= \Big(\sum_{d\neq d_i} e^{-(c(d,x)-c(d_i,x))^2/\sigma^2}\Big)^{-1} \\
C(x) &= \frac{1}{N}\sum_{i=1}^{N} C'(x,d_i)
\end{aligned}
\tag{3}
$$

2.2 Determination of the filtered region

To avoid introducing extra artifacts and improve the ability of real-time processing, the filtered region is determined firstly. Generally, the significant noise occurs around the object's edges according to the empirical measurement. In this paper, we assume that discontinuities in depth maps is the center line of the filtered region. Canny edge detector is applied to the gray-level depth image to obtain the location of the filtered region. Then, object's edge is dilated using 3×3 rectangular structuring element. To facilitate further processing using the method proposed in this paper, the pixels in the filtered region are labeled.

2.3 Approximation of the CWDF

According to Eq. 1, it is also signal-related filter, including $f_r(\|\tilde{I}(p)-\tilde{I}(q)\|)$ and $f_{r'}(\|I(p)-I(q)\|)$ terms. To employ the recursive implementation of the Gaussian filter, The CWDF is approximated to approach the evolutional expression as shown as follows.

$$
\begin{aligned}
I^C(p) =& \frac{\alpha(\triangle\Omega)}{K_1(p)}\sum_{q\in\Omega} f_s(\|p-q\|) f_r(\|\tilde{I}(p)-\tilde{I}(q)\|) I(q) \\
&+\frac{1-\alpha(\triangle\Omega)}{K_2(p)}\sum_{q\in\Omega} f_s(\|p-q\|)* \\
&C(q) f_{r'}(\|I(p)-I(q)\|) I(q)
\end{aligned}
\tag{4}
$$

where $K_1(p)$ and $K_2(p)$ are normalizing factors. The proposed filter is decomposed into a joint bilateral filter and a standard bilateral filter related to confidence map $C(q)$. The CWDF can be expressed as Eq. 5. $B^J_{I(p)}(p)$ and $B^S_{I(p)}(p)$ are defined for the joint bilateral filter and standard bilateral filter respectively. In practice, $min(I)$ and $max(I)$ in two bilateral filters are usually different, so we set $[min(min(I),min(\tilde{I})),max(max(I),max(\tilde{I}))]$ as the range of intensity value in order to guarantee the completeness of LUT.

$$
\begin{aligned}
I^C(p) =& CW_{I(p)}(p) \\
=& \alpha(\triangle\Omega) B^J_{I(p)}(p) + (1-\alpha(\triangle\Omega)) B^S_{I(p)}(p)
\end{aligned}
\tag{5}
$$

Only N' intensity values are chosen from $[min(min(I),min(\tilde{I})),max(max(I),max(\tilde{I}))]$ to be implemented the proposed filter. The remaining can be obtained by the linear interpolation after looking up the both the nearest intensity values from LUT. For instance, for $I(p)\in[I(p_1),I(p_2)]$,

$$\begin{aligned} I^C(p) =& (I(p_2) - I(p))CW_{I(p_1)}(p) \\ &+ (I(p) - I(p_1))CW_{I(p_2)}(p) \end{aligned} \tag{6}$$

In addition, to ensure the synchronous implementation of the proposed method, the capturing and filtering procedure are run in two separate threads in a thread-safe way based on interlock mechanism, which prevents more than one thread from using the same variable simultaneously. And in practice, the filter is implemented on a Nvidia's graphics card to achieve the real-time denoising performance. As a data-parallel computing device, the massive stream processing is executed using a high number of threads in parallel. In the experiments below, the CUDA programming framework [12] is employed to port the filter processing onto graphics hardware only. And the selection of the appropriate granularity is a trade-off between the runtime and memory.

3 Experimental results

In this section, Kinect from Microsoft is chosen to evaluate the proposed algorithm for its simple setup and depth map at a granularity of 640×480 pixels, which is higher than 128×128 of Flash Lidar and 176×144 of Mesa Swissranger^{TM}. In addition, the resolution of color image is the same with the depth map, so that we can align them easily. Especially, both of them can record scenes at up to 30 fps in real time.

To demonstrate the effectiveness and accuracy of the proposed method, the algorithm was applied to the real-world sequences and scenes from the Middlebury stereo benchmark data set [13] and Advanced Three-dimensional Television System Technologies (ATTEST) [14]. In the following, two main aspects are discussed in details, including the resultant comparison with the joint bilateral filter and noise-aware filter in [9]. All computations in the experiments were performed on two Intel Core (TM) P8700 CPUs with an Nvidia GeForce GT 120M.

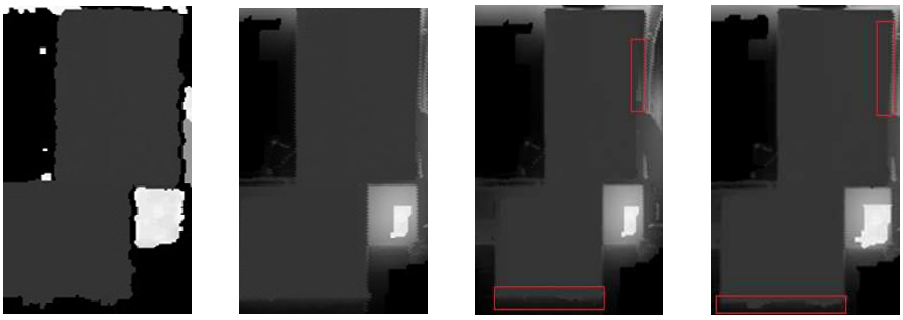
3.1 Quality and effectiveness analysis

Several different scenes were captured to test the proposed algorithm. One scene shows two boxes with severely noise-affected edges, as shown in Figure. 1(b) in the red rectangular region. Particularly, in Figure. 4(a), the color image is modified by drawing a red rectangle across the edge of one box to evaluate the performance of the joint bilateral filter specifically.

Figure. 2 shows the results using the proposed filter, comparing with that using joint bilateral filter and noise-aware filter. As visible, all algorithms above can reduce the noise level. However, the intensity texture pattern of the red mark in the color image is introduced as 3D structure into the filtered depth by the joint bilateral filter, as shown in right red rectangle in Figure. 2(c). Although texture copying is avoided using the noise-aware filter, there are still some noise around the bottom edge due to the low confidence measurement, shown in Figure. 2(d). In contrast, confidence weighted depth filter gets the edges across the boxes straight, and prevents the texture copying effectively. Therefore, the proposed method can sharp details in the original depth image and improve the reconstruction quality.

(a) Color image (b) Gray-scale depth image

Fig. 1. Color image and depth image with two boxes.



(a) Depth image (zoomed) (b) Proposed method (c) Joint bilateral filter (d) Noise-aware filter

Fig. 2. Result comparison using the different methods.

The detailed view warped by color information in Figure. 3. The noise across the edges is almost removed, while the depth measurements are severely interfered in the original 3D view. The point cloud contains more than $300,000$ points, although some points with invalid depth are eliminated. Due to the average filtering time $47ms$ with GPU support, the capturing rate of the device is set to 20 fps. Then the resultant colored depth map is transmitted to a server for 3D reconstruction using SURF (Speeded Up Robust Feature) feature descriptor.

3.2 Quantitative accuracy

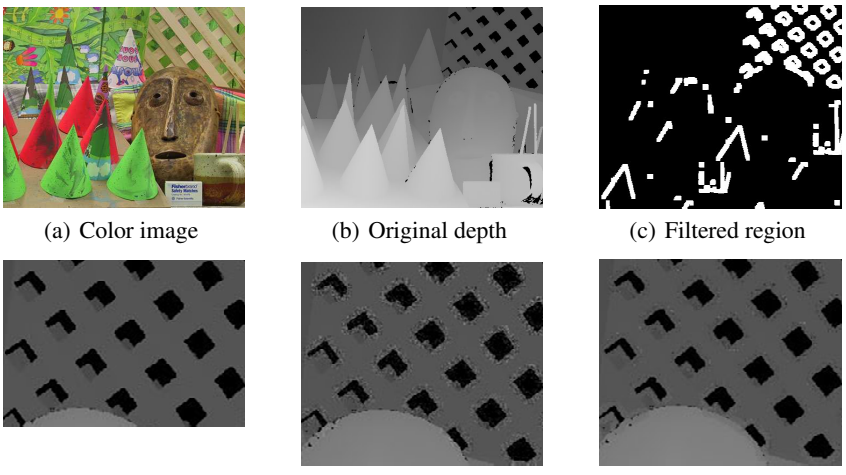
To evaluate the performance of our filter, Cones scene from Middlebury stereo data set and video-plus-depth sequence "Interview" from ATTEST are considered. In order to illustrate the effectiveness of the confidence estimation and quantitative accuracy, confidence based noise is superimposed on the original depth image, which is considered as the ground truth. Then the filtered depth map will be compared with the ground truth. Supposed the confidence measurement C depends on the distance from the position of current pixel p to the image center c, and the noise n is subject to Gaussian distribution with mean 0, the noise model is defined as follows.

(a) Original view (b) View after filtering

Fig. 3. Comparison of the 3D view.

$$C = 1 - \frac{\|p-c\|}{w}$$
$$n \sim N(0, 100 \cdot (1-C)) \tag{7}$$

where w is a factor to balance the influence of the additional noise. Figure. 4(e) shows the depth image with Gaussian noise, where the image distortion only exists across the edge, as shown in Figure. 4(d).



(a) Color image (b) Original depth (c) Filtered region

(d) Original depth (e) Noised image (f) Filtered result

Fig. 4. Algorithm evaluation using the cones scene from the Middlebury data set.

To evaluate the numerical accuracy, peak signal-to-noise ratio (PSNR) is employed. Given two gray-scale images I, I', usually the ratio is defined using the mean squared error (MSE).

$$MSE = \frac{1}{hw}\sum_{i=0}^{h-1}\sum_{j=0}^{w-1}\|I(i,j) - I'(i,j)\|^2$$
$$PSNR = 10 \cdot log_{10}(\frac{MAX^2}{MSE}) \tag{8}$$

where h and w is the height and width of input image, and MAX is the maximum of the pixel value, which means $MAX = 255$, if both input are gray images, where the pixel is indicated by 8 bits. The PSNR of the depth with noise and original depth image is 24.16 dB. To speed up the implementation, only N' intensity values are selected to perform the proposed algorithm. Figure. 5 shows the result of the PSNR accuracy using different number of intensity levels. When 8 levels are used, the noise level is reduced greatly, and an acceptable PSNR value is achieved, although there exists visible difference. Figure. 4(f) shows the filtered result using 8 intensity levels. Note that as the level number increases, we have to compromise the computational time.

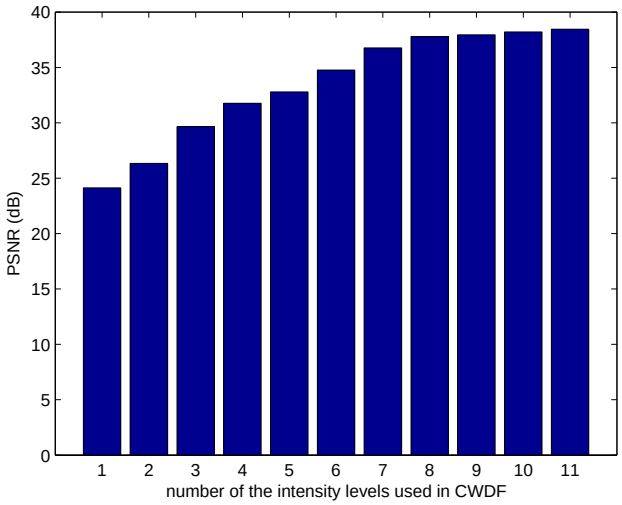


Fig. 5. PSNR accuracy using different number of intensity values.

Moreover, video-plus-depth sequence “Interview” from ATTEST is used with the yuv420 video format for comprehensive test. The sequence is decoded based on the leading audio/video codec library in FFmpeg. Every frame is in the size of 720 by 576 for both color and depth sequence. The depth is indicated by 8 bits, the brighter, the closer. The first 60 frames are chosen to implement our method, joint bilateral filter and noise-aware filter respectively. Similarly, noise superimposition is repeated based on the same noise model above. Then the PSNR accuracy of each frame is calculated, as shown in Figure. 6. the proposed method has the best performance with 39 dB on

average. Due to the resolution gets bigger, the GPU implementation is prolonged to the averaging 55 *ms*.

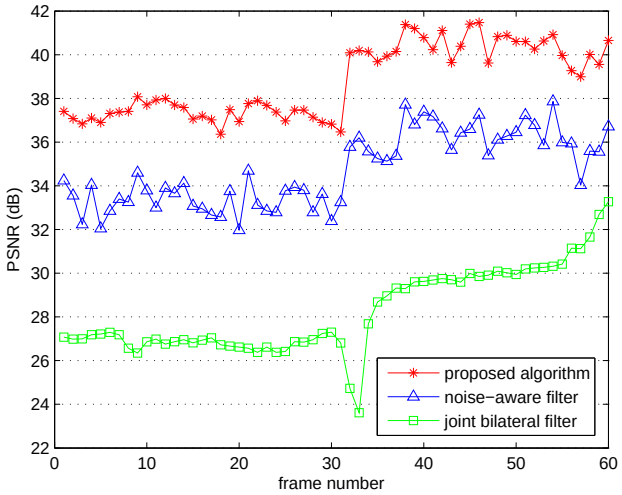


Fig. 6. PSNR accuracy with respect to different methods.

4 Conclusion

The accurate 3D reconstruction is required for the environmental perception and rehabilitation process. This paper presented a confidence estimation based depth filter to recover the edges around the object in 3D reconstruction of a scene with color and depth images. The proposed method takes into account both the inherent depth nature of real-time depth data and the color information from video camera to reconstruct a multi-lateral filter. Compared with the existing joint bilateral filter and noise-aware filter, the experimental results show that the proposed method obtains more detailed information around the edge of an object and reduces the geometric distortions. Using the parallel data processing based on GPU implementation, the real-time high-quality 3D reconstruction is achieved.

In future, the adaptive parameter selection for the kernel size in the range term will be achieved, while these parameters are set manually according to the different scenes in current phase. In addition, multiple depth sensors will be employed for dynamic environment. A corresponding distributed system will be set up, and more studies will concentrate on the time synchronization of multiple data streams in order to ensure the temporal stability.

References

1. Natalia Díaz Rodríguez, Robin Wikström, Johan Lilius, Manuel Pegalajar Cuéllar, and Miguel Delgado Calvo Flores. *Ubiquitous Computing and Ambient Intelligence. Context-Awareness and Context-Driven Interaction: 7th International Conference, UCAmI 2013, Carrillo, Costa Rica, December 2-6, 2013, Proceedings*, chapter Understanding Movement and Interaction: An Ontology for Kinect-Based 3D Depth Sensors, pages 254–261. 2013.
2. Dan Mircea Suciu, Bogdan Andrei Pop, Rares Urdea, and Bogdan Mursa. *On the Move to Meaningful Internet Systems: OTM 2014 Workshops: Confederated International Workshops: OTM Academy, OTM Industry Case Studies Program, C&TC, EI2N, INBAST, ISDE, META4eS, MSC and OnToContent 2014, Amantea, Italy, October 27-31, 2014. Proceedings*, chapter Non-intrusive Tongue Tracking and Its Applicability in Post-stroke Rehabilitation, pages 504–513. 2014.
3. Tien Tuan Dao, Philippe Pouletaut, Didier Gamet, and Marie Christine Ho Ba Tho. *Knowledge and Systems Engineering: Proceedings of the Sixth International Conference KSE 2014*, chapter Real-Time Rehabilitation System of Systems for Monitoring the Biomechanical Feedbacks of the Musculoskeletal System, pages 553–565. 2015.
4. Farzin Amzajerdian, Diego Pierrottet, Larry Petway, Glenn Hines, and Vincent Roback. Lidar systems for precision navigation and safe landing on planetary bodies. In *Proceedings of SPIE*, volume 8192, 2011.
5. C. Dal Mutto, P. Zanuttigh, and G.M. Cortelazzo. Accurate 3D Reconstruction by Stereo and ToF Data Fusion. *GTTI,(Brescia, Italy), June*, 2010.
6. Roanna Lun and Wenbing Zhao. A survey of applications and human motion recognition with microsoft kinect. *International Journal of Pattern Recognition and Artificial Intelligence*, 29(05):1555008, 2015.
7. Pierre Kornprobst, Jack Tumblin, and Fr do Durand. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*, 4(1):1–74, 2009.
8. Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. *ACM Trans. Graph.*, 26, July 2007.
9. Derek Chan, Hylke Buisman, Christian Theobalt, and Sebastian Thrun. A noise-aware filter for real-time depth upsampling. In *In Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, 2008.
10. B. Huhle, T. Schairer, P. Jenke, and W. Strasser. Robust non-local denoising of colored depth data. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on*, pages 1 –7, june 2008.
11. Sing Bing Kang, R. Szeliski, and Jinxiang Chai. Handling occlusions in dense multi-view stereo. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–103 – I–110 vol.1, 2001.
12. Nvidia. http://developer.nvidia.com/cuda-toolkit-40. 2011.
13. Middlebury. http://vision.middlebury.edu/stereo/data/scenes2003/. 2003.
14. André Redert, Marc Op de Beeck, Christoph Fehn, Wijnand IJsselsteijn, Marc Pollefeys, Luc J. Van Gool, Eyal Ofek, Ian Sexton, and Philip Surman. Attest: Advanced three-dimensional television system technologies. In *3DPVT*, pages 313–319, 2002.